Improving Performance In Mobile Ad-hoc Network Using Data Caching Schemes: A Review

Ms. Mandhare V.V., Dr. R.C. Thool
Research Scholar, Professor
v_mandhare@yahoo.com, rcthool@yahoo.com

Abstract

Abstract—Mobile Ad-Hoc Network (MANET) is self organizing infrastructureless network. Performance of protocol is mainly depending on how optimally route is discovered. On demand routing protocol faces many problems as route failure, cache management, cache updation and cache coherence. This paper focus mainly on cache problem faced by mobile ad-hoc network and surveys the new data caching schemes to improve the performance of protocols. Some assumption has been made to enhance performance of protocols by enhancing caching scheme.

1. Introduction

Mobile ad-hoc network (MANET) [1] is collection of nodes which communicates with each other without infrastructure. The two main characteristics of MANETs are mobility and multi hop communication. MANET’s applications include military scenarios, sensor networks, rescue operations etc. In these types of networks, devices generally have limited energy reserves and processing capabilities. Bandwidth is also a scarce resource, limited by the nature of the wireless medium. In a data-management point of view, these restrictions introduce several issues that need to be addressed. Data transfers must be reduced and mechanisms must be deployed in order to confront the frequent disconnections and low bandwidth constraints. Therefore it is a challenging task to present the data efficiently by reducing the delay or waiting time to the end user.

Ad-hoc network must optimally use scarce resources such as bandwidth, computing power, memory and battery power. Cache memory is treated as one of the possible techniques that can enhance networks performance. Caching is widely exploited in MANETs for query processing with minimum access latency.

2. Caching

Cache provides temporary storage of path that have been requested by an application. If path is discover more than once instead of reconstructing path again it is requested from cache for further processing. It ultimately improves the performance of network by reducing the time required to get a requested path. If requested data is contained in the cache (cache hit), this request can be served by simply reading the cache, which is comparatively faster. Otherwise (caches miss), the data has to be recomputed or fetched from its original storage location, which is comparatively slower. Hence, the greater the number of requests that can be served from the cache, the faster the overall system performance becomes.

In wireless Ad-hoc networks, the data caching strategy becomes a problem. Caching data at nodes helps a wireless network system to run faster, reduce the cost of bandwidth and avoid overload at nodes with more efficiently. After it has been decided to cache, the next issue is what to cache and where [2].
3. Cache Strategies

In this topic we see how the structure of cache in on-demand protocol is present.

I. Cache Structure:

Cache Strategies is very important for developing route in on-demand routing protocol. In the cache structure, there are two kinds of caches are used. First, a path cache which represents a complete path (a sequence of links), that caches the routes separately. Second, a link cache that represents when a node caches each link individually, adding it to a graph of links. In case of the path cache structure when the source node A adds a new route A-B-F-G-H to its cache, it has to add the whole path as independent entry in its cache. While in case of the link cache when the source node A adds a new route A-B-F-G-H to its cache, it has to add the link B-F only, since rest of the links already exist in its cache. However, in path cache, it does not effectively utilize all of the potential information that a node learns about the state of the network and there is no sharing in the data structure even when two paths share a number of common links as stated in [3, 21]. While in link cache, it has a conventional graph data structure, in which, all the links are learned from different route discoveries or from the header of any overhead packets can be combined together to form a new routes in the route cache of the network. Therefore, this is not possible in path cache due to the separation of each individual path in the cache. In addition, several studies have been shown that link cache outperforms path cache when the network load is high as examined by [15, 20, 21-23]. Because of the link cache can delete only a broken link when the link causes a path to break. While in path cache whole the route which link failure occurred will be deleted.

II. Cache Capacity:

According to Hu and Johnson [3] cache capacity is also important cache for choice in designing a cache strategy for on-demand protocol. For link cache, it can store all the links that are discovered in the cache, since there is fixed maximum N x N links that may exist in an ad-hoc network with N nodes; these stored routes require less storage space. While in the path cache, the maximum storage space is required much larger.

Since a path is stored separately. Cache capacity can be divided into two halves: first half-called “Primary Cache”, which represents paths that have used by the current node. Second, half-called “Secondary Cache”, which represents paths that have not yet used. When the source node attempts to add new path that have learned and not yet been used into the cache, old paths in the secondary cache are removed due to limitation of the cache capacity in the network. While with primary cache, old paths are removed more actively due to the operation of route maintenance.
III. Cache Timeout:

Cache timeout policy is required to be implemented in link cache due to time varying topology of the ad-hoc network caused by node’s mobility that can cause stale routes in the cache over period of time as stated in [3, 5]. Deriving proper cache timeout policy is important role for ensuring cache freshness. Cache timeout policy in link cache gives a timeout that may be static or adaptive to remove the stale routes from the cache. Stale routes is a big issue in link cache structure, where individual links are combined together to find out best path between source and destination. However, cache timeout policy is not possible to setup timeout in path cache structure due to the limitation capacity of the storage space. According to Hu and Johnson [3] there are two kinds of cache timeout are used: static timeout approach and adaptive timeout approach. Static timeout is assigned the same timeout value for every link cache entry, each link is removed from its cache after specific value of time has elapsed since the link was added to the cache. In contrast, adaptive timeout is assigned a timeout value based on the stability of the link endpoints. The timeout can be calculated based on the elapsed time since the link was last used and the last time the link was observed.

4. Cache Policies

A cache policy defines rules that are used to determine whether a request can be satisfied using a cached copy of the requested resource. Applications specify client cache requirements for freshness, but the effective cache policy is determined by the client cache requirements, the server's content expiration requirements, and the server's revalidation requirements. The interaction of client cache policy and server requirements always results in the most conservative cache policy, to help ensure that the freshest content is returned to the client application.

Cache policies are

1. Read policy: In this type of policy data is read from the cache as request is send from the protocol.

2. Load policy: In this type of policy data or information about route is load into cache.

3. Replacements policy: In cache replacement strategies if cache found is to be full and required extra space to store in data cache at that time some unused information is deleted from the cache and new route information or data is replace with the new information that is required.

4. Fetch policy: In this type of policy data is fetch into main memory of the cache.

5. Write policy: This type of policy writes the data into cache structure where the space is present if space is unavailable then it is unable to write into cache.

5. System Architecture for cache scheme

Caching scheme provides the data management service to the upper layer in MANET environment. For perfecting purpose network traffic information which is present in the data link layer is retrieved by middleware layer.
**Application layer:** It is providing service to the users for interacting to application or networking services. Application layer uses HTTP, TELNET, FTP, TFTP.

**Middleware layer:** It provides service to application layer. It gives service as shared memory, group communication and service location. Middleware consist of following various block.

- **Cache management:** Cache management consist of cache admission control, cache consistency maintenance and cache replacement.
- **Cache consistency:** It keeps the cached data items synchronized with the original data items in the data source.
- **Cache admission control:** A node will cache all received data items. After the cache space becomes full, the received data item will not be cached if the data item has a copy within the cluster.

**Cache replacement:** If new data item is arrived for caching and there is no space in then cache replacement algorithm is used to store cache data items. LRU, LRU_MIN are cache replacement algorithm.

**Cache consistency:** It keeps the cached data items synchronized with the original data items in the data source.

**Information search:** Deals with locating and fetching the data item requested by the client.

**Prefetching:** Responsible for determining the data item to be Perfected from the Data Centre for future use.

**Transport layer:** It is responsible for providing data delivery transportation between the applications in the network by using the protocols like TCP, UDP. It includes the functionalities like Identifying the services, segmentation, sequencing and reassembling and error correction.

**Network layer:** It is responsible for providing logical addressing and path determination (routing). The routing protocols such as AODV, DSR, DYMO, etc. are responsible for performing path determination (routing).

**Data link Layer:** Provides apparent network services so that network layer can be ignorant about the network topology and provides access to physical networking media. It includes error checking and flow control mechanism.

---

Fig: System Architecture Of Cache Scheme
6. Cache replacement policy for wireless network

Cache replacement policy for wireless network is divided into following categories.

1. Function based cache replacement:
   In this cache replacement each data item is assigned a value using a value function by combining parameters. Parameter considers are data access pattern, access costs, mobility pattern, connectivity, bandwidth, update rates, location dependence of the data. Numbers of different cache eviction algorithm are used in wireless network.

2. Location based cache replacement:
   Location dependent cache replacement depends on location and differs as the users move from one location to another. Parameters consider in this policy are area, distance and direction of user. Distance is distance between mobile users from its current location.

3. Semantic cache replacement:
   In semantic caching each mobile client have a semantic description of data and query processing make use of this data. Semantic caching is ideal for location dependent services as the probability of having similar queries is high. There may be a large degree of overlap in the results of consecutive queries.

7. Cache replacement strategies in Ad-hoc network

Available cache replacement algorithm improves the performance of cache by selecting subset of data for caching.

1. Uncoordinated cache replacement: When cache is full then decision is made only considering the local parameter. Different cache replacement strategies used under this category are LRU (Least Recently Used), LRU min, LUV (Least Utility Value) etc.

2. Coordinated cache replacement:
   Coordinated cache replacement strategy for cooperative caching schemes in mobile environments should ideally consider cache admission control policy. Cache admission control decides whether the incoming data is cacheable or not. Cache space can be saved by admission control, thereby evicting more data. Evicted data is stored in neighboring node.

   Numbers of different cache replacement policy are TDS, Energy efficient cooperative cache replacement problem (ECORP), count vector etc.

8. Issues in caching

1. No provision has to be taken by on-demand protocol to avoid flooding for new route to destination. For getting the route, DSR has flood route and store it into cache.

2. When mobility is high, entries in route caches become invalid or ineffective. Data packets suffer needless delays, when an ineffective route is used. When an invalid route is getting route failures will generate flooding, creating latency for data packets.

3. Stale route problem in cache is the major problem in on-demand protocol. Hence pre-active and post-active are the key sources in cache staleness.

4. There is incomplete error notification present when the route or link is broken. The information is not feed to all caches. Route error is unicast to the source whose data packet is accountable for identifying the link breakage by means of a link layer.
feedback. Therefore few caches are cleaned. Many caches become as it is successive route request from source.

5. Also there is no mechanism for expiry stale route. Stale cache entry is present in the cache if it is not clean by error mechanism.

6. There is no technique to resolve the freshness of any route information. Quick pollution is main problem in on-demand protocol. As there will be a large number of routed data packets upstream carrying the stale route to unerase the route. The above possibility increases at high data rates. By the moderate use of snooping, this difficulty is compounded. Hence cache pollution can transmit quite rapidly.

9. Literature survey

Cooperative caching in mobile ad-hoc networks aims at improving the efficiency of information access by reducing access latency and bandwidth usage. Cache replacement policy plays a vital role in improving the performance of a cache in a mobile node since it has limited memory. Joy P.T, Jacob K. P [8] proposes a new key based cache replacement policy called E-LRU for cooperative caching in ad-hoc networks. This scheme for replacement considers the time interval between the recent references, size and consistency as key factors for replacement. The algorithm takes in to account the inter arrival time of recent references, size and consistency for page replacement. In LRU only the last time of reference is taken and the numbers of references are not considered. Since the inter arrival time of the recent reference is taken more preference is given to objects that have been accessed more than once. Hence we are able to distinguish between data that are frequently referenced with that of occasionally referenced. As we are not caching data with bigger size cache space can be saved. Since the algorithm is based on a key based approach it is simple to implement.

Fenglien Lee, Carl T. Swanson, Jigang Liu [9] proposes an efficient algorithm for route discovery and management, and mobility handling for on-demand cache routing on mobile ad-hoc networks (MANET). L-1 and L-2 route apply on caches in each node to manage this algorithm efficiently. For route discovery, simpler and more efficient broadcast approach than in AODV (Ad-hoc On-demand Distance Vector Routing). For route maintenance, the LRU replacement policy in caches to maintain route table and to remove the time-to-live parameter in some on-demand protocols. LRU replacement features in L-1 and L-2 caches for route table creation, updating, and maintenance. The OCR algorithm with double-level route caches solved most problems in on-demand routing, such as route tables in “slow” main memory, long connection setup delay, broken link repairing, huge routing overhead for long routes, lengthy data packet in source routing, sending beacons (“hello packets”) periodically, control overhead for complicated IDs in data packets, to setup TTL (time-to-live) in a packet or a route path, and to update the stale routes in the route table or cache frequently.

Dimitrios Koutsonikolas Saumitra M. Das Himabindu Pucha Y. Charlie Hu [10] Flooding of requests has used a TTL (time-to-live) large enough to reach all nodes in the network to ensure successful route discovery in one round of flooding. It was shown that the generic minimal cost flooding search problem can be solved via a sequence of flooding with an optimally chosen set of TTLs. The theoretical result, when applied to DSR route discovery, does not take into account optimizations such as route caching and overhearing, which can significantly reduce the frequency and the propagation range of route discovery operations. Equally importantly, the
impact of using a sequence of flooding on the packet delivery delay is not clear. The impact of using the optimal TTL sequence-based route discovery on DSR routing performance. Results show when caching and overhearing are considered, the route discovery enhanced by an optimal TTL sequence has very similar overhead but higher delay than the basic route discovery mechanism.

Narinderjeet Kaur, Maninder Singh [14] In order to avoid route discovery mechanism each time when the packet is transmitted, the route caching technique is used. Route caching is the major approach to decrease the flooding of the network by avoiding the route discovery operation as much as possible. Because the frequently use of route discovery mechanism is very costly in terms of delay and bandwidth consumption which can cause congestion and long delay. Therefore the efficient caching strategies have the great impact on the performance of the on-demand routing protocol. In this the effect of the cache expiry time and cache capacity on the performance of the on-demand routing protocol. The simulation results indicate that the cache capacity can indirectly affect the performance of the routing protocol. The performance of the on-demand protocol decreases if we increase the cache size, because increase in cache size cause stale routes in the cache. The stale routes are generated due to the mobility of the nodes. To reduce or remove the stale routes cache expiry time is used.

To avoid performing route discovery mechanism before each data packet is sent, DSR needs to formerly cache the routes discovered. T.C. Huang et al. proposed two mechanisms to improve cache performance of DSR, RERR-Enhance mechanism and hierarchical link cache structure as described in [12], the average end to end delays and routing overhead is improved with these mechanisms.

The route cache is used in DSR protocol to store all the routes are learned from the source node and to avoid unnecessary route discovery process. Naseer Ali Husieen et al. described in [14] with high mobility situations and high load network traffic stale routes will be generated. These stale routes can mainly affect the performance of DSR protocol which cause long delay, increase the packet loss, increase the overhead and reduce the efficiency of protocol. Therefore efficient mechanism for updating the route in the cache of DSR protocol is needed.

Shukla et al. [13] proposed new route cache maintenance in DSR protocol in order to improve the performance of the DSR protocol. This was achieved by allowing nodes to learn about the route caches of neighbors’ nodes, instead of sending route error packets to the source, due to link failure, before forwarding the packet to the destination.

R. Bhuvaneswari, M. Viswanathan [15] reduces the effect of overhearing and avoids the stale route problems while improving the energy efficiency using the Efficient Source Routing Scheme (ESRS) algorithm. Due to the lack of route cache update, the stale route entry and overhearing is originated among the network. For that, we developed five mechanisms to improve route cache performance in DSR.

S. Menaka and M.K. Jayanthi [16] Routes are maintained in route cache by every node as they learn from forwarding data packets. While route cache has their own benefits of reducing the overhead in the network, it possess inefficiencies due to stale paths and use of low quality paths when feasible paths become available. Based on the studies of performance of DSR, a new approach to update the route cache to remove the stale paths effectively using preemption in route switching by predicting the possibility of current link breakage by sensing the RSSI value.

Hasan Artail, Haidar Safa, Zahy Abou-Atme, Nabeel Suleiman [17] Introduces nodes that cache submitted query. How the system is formed and how the requested data is found if cached or retrieved from the external database and then cached. For enhancement of system’s performance and reliability cache invalidation, cache
replication, partial cache reuse, and the graceful shut down of devices.

Improve cache effectiveness and packet overhead by discovering new technique as smart packet [18]. Using this technique it is observe that invalid cache entries decreased with density of network.

To avoid route discovery mechanism in DSR needs mainly to cache the route discovers T.C. Huang et al. proposed two mechanisms to improve cache performance of DSR: RERR-Enhance mechanism and hierarchical link cache structure as described in [18], the average end to end delays and routing overhead is improved with these mechanisms.

Shobha K.R. et al. [19] presents an analysis of the effect of intelligent caching in non clustered network, using on demand routing protocol. In Intelligent Caching technique a node not only saves the path discovered during route discovery for itself but also for others who are located close to it. This technique reduces the number of route request packets unnecessarily circulating in the network, when the path it requires is present in its neighborhood. This technique drastically increase the available memory for caching the routes discovered without affecting the performance of the DSR protocol.

10. Research Challenges

In recent age we are facing many problems as route failure, dynamic topology, battery power, mobile node etc. Due to the dynamic nature of the node it is very difficult to manage small network. Therefore there are many challenges face by caching scheme such as cache replacement, cache coherence, replication of data, power consumption, cache expiration time, cache size problem. Hence to avoid such cache problem new caching scheme must be present to improve the performance of the network using any type of protocol.

Conclusion

In this paper we introduce the cache issues for ad-hoc network. Problems face like replication of data, power consumption, and cache replacement, stale route problem in caching technique intelligent caching scheme is required to improve the performance of network. We provide some literature review for cache replacement and stale route problem in ad-hoc network. Also we are providing some future direction to enhance networks performance.

References


